Forest Fire Recognition Based on Lightweight Convolutional Neural Network

Zhixiang Li, Hongbin Jiang, Qixiang Mei, Zhao Li*

College of Mathematics and Computer, Guangdong Ocean University, China
839069031@qq.com, 1434709585@qq.com, nupf@163.com, zhaoli@gdou.edu.cn

Abstract

In recent years, there have been numerous forest fires, and fire identification technology has become increasingly influential in both academic and industrial fields. At present, most automatic fire alarm systems are limited to identification by sensors such as temperature, smoke, and infrared optics. One of the existing solutions is the method of image feature extraction, which does not need to rely on specific sensors and can be easily embedded in different devices. However, this method has the disadvantage that it is difficult to extract features from image data. To attack this issue, this paper proposes a lightweight convolutional neural network for forest fire recognition. Firstly, three-channel color images of three scenes are constructed as the input of the convolutional neural network, and the initial data are pre-processed and enhanced. Secondly, a deep convolutional neural network with multiple layers of convolution and pooling layers is constructed. Finally, the Softmax function is used to classify the fire recognition scenes. The experimental results show that our approach outperforms these selected techniques in the effectiveness and accuracy.

Keywords: Forest fire recognition, CNN, MobileNet

1 Introduction

Since the birth of forests on Earth, forest fires have not stopped. As long as the forest is not cut down, there is the possibility of fire. In terms of natural conditions, with the rapid development of economy, the emission of greenhouse gases has gradually increased, and the situation of global warming has gradually worse. Fires are occurring more frequently on a large scale in countries around the world, such as the rare forest fires near Athens, Greece, in the summer of 2018; and the frequent Amazon Forest fires in 2019, with the Amazon region, where Brazil and Peru meet Bolivia, being hardest hit by fires. Statistics from the Brazilian National Institute for Space Research show that from the beginning of 2019 to the end of August, there were 44,058 fires in the Amazon, an increase of 145% over 2018. Of the 300 million tonnes of carbon dioxide emitted each year, 200 million tons came from fires in the Amazon rainforest [1]. In March 2020, a forest fire broke out in Xichang City, Liangshan Prefecture, Sichuan Province, causing immeasurable resource, environmental and ecological losses. Therefore, the monitoring of forest fires is particularly important. At present, many kinds of sensors are used to monitor fires. For example, a ZigBee-based wireless sensor is proposed by Huang He of Hunan University of Technology and Business [2]. Qin Yulin of Southwest Jiaotong University and others proposed a forestry fire monitoring and early warning system based on NB-IoT narrowband communication technology and wireless sensor network technology [3]. Wang Wei et al. used radar data, combined with the temperature, precipitation, relative humidity, wind speed and other information obtained by meteorological sensors, and established a radar-based multi-source information forest fire monitoring model through data processing in the information layer, feature layer, and decision-making layer. [4]. There are common and unavoidable drawbacks in using sensors for fire detection. Its environmental adaptability is poor, and slight environmental changes will affect the recognition accuracy. In addition, changes in the physical properties of the sensor itself may also affect the recognition results.

In recent years, with the development of machine learning and image recognition technology, detection technology based on picture and video recognition has been widely used in forest fire prevention. At present, the common image detection method is the recognition of smoke images and flame images. Fire smoke exists in various time periods of fire, and smoke can be analyzed from various aspects and perspectives to achieve fire prevention and detection. For example, Yang Jian et al. [5], Du Jiaxin et al. combined DBN-CNN to propose a forest fire smoke identification algorithm [6]; Zhu Lei proposed a forest fire smoke detection algorithm based on image enhancement and multi-features [7]. Flame image is one of the features that can most directly reflect the occurrence of fire, so there are many researches on the recognition technology of flame image in recent years. A flame recognition algorithm based on graph features combined with LPQ histogram features [8]. Zhu et al. proposed a lightweight network flame smoke detection algorithm, which using a lightweight neural network MobileNetV3 replaces the original backbone feature extraction network of YOLOv4. Based on the above results, it can be seen that the method of machine learning with neural network has a good effect in the field of fire recognition.

This paper aims to apply a lightweight neural network built by MobileNet to a system that combines smoke image recognition and flame image recognition, and judges the fire level according to the classification results, so as to achieve prevention and detection. Different from traditional technical methods, neural network can automatically complete the feature extraction of flame and smoke in the early stage of fire,

*Corresponding Author: Zhao Li; E-mail: zhaoli@gdou.edu.cn
DOI: 10.53106/160792642022092305023

thus avoiding the complexity and blindness of manual feature extraction. Using lightweight neural networks, the number of parameters can be effectively reduced through optimization strategies such as multiple convolution kernels, batch normalization, and custom loss functions. Thereby, the problem that the traditional convolutional neural network has a large memory requirement and a large amount of computation, which cannot be arranged on mobile devices and embedded devices, is solved. The combination of smoke and flame improves the versatility and applicability of the system, and provides technical support for forest fire prevention and identification.

The main contributions of this paper are summarized as follows: (1) Based on MobileNet, we established a forest fire recognition system [9]; (2) We compared MobileNet with several existing mainstream convolutional neural networks in the field of forest fire recognition; (3) We constructed a dataset of forest fires, which includes 3 subsets.

2 Our Techniques

2.1 Datasets Loading

We downloaded data from existing public datasets to construct the dataset for smoke and flame recognition. The dataset is enhanced through content transformation and scale transformation, and finally 5360 images were obtained. The quantity of experimental dataset is shown in Table 1.

Table 1. Datasets

Name of subset	Quantity of training set	Quantity of validation set	Total
Fire	2060	515	2575
Smoke	1117	279	1396
Fog	1112	277	1389

2.2 Datasets Processing

The processing of the datasets before the experiment starts was very necessary, the image data used in the neural network needed to have as many kinds as possible. In this study, after the data collection, the collected images were classified and labeled, and then divided into three categories: Fire, Smoke and Fog. This paper adjusted the size of all images to 224 * 224 * 3 when loading data and normalized the images when training the models. The data which between 0 and 255 will be processed to -1 to 1 (MobileNet). The ratio of training set to validation set was 8:2. The selection of training set and validation set images was based on the principle of random seeds, and normalized processing was adopted.

Part of the data in the dataset are shown in Figure 1, Figure 2 and Figure 3.

Figure 1. Part of the dataset on Fog

Figure 2. Part of the dataset on Fire

Figure 3. Part of the dataset on Smoke

2.3 Convolutional Neural Networks

Convolutional Neural Networks (CNN) is a process of obtaining different classification models by means of automatic learning through neural network backpropagation. Deep learning is a hierarchical machine learning method that includes multi-level nonlinear transformation. Deep neural network is the main form at present. The connection pattern between neurons is inspired by the organization of animal visual cortex, and convolutional neural network is one of the classic and widely used structures. The characteristics of local connection, weight sharing, and pooling operation of convolutional neural network can effectively reduce the complexity of the network, reduce the number of training parameters, and make the model have a certain degree of invariance to translation, distortion, and scaling. It has strong robustness and fault tolerance, and is also easy to train and optimize. Based on these superior properties, it outperforms standard fully connected neural networks [10] in various signal and information processing tasks. There are various applications of neural networks at present [11-13]. For example, Liu Wanjun et al. identified traffic signs based on multi-scale convolutional neural networks [14], and Wu Wenjuan et al. proposed a new lightweight face recognition convolutional neural network -- Emfacenet. The high precision and the significantly improved operating efficiency can be applied to embedded and other hardware resource-constrained fields to realize face recognition [15], and Yue Youjun et al. used the improved Cascade RCNN network to target the tomato in the greenhouse. Experiments have obtained better detection results [16].

The convolutional neural network mainly includes convolution and pooling operations, and automatically learns the characteristics of images at various levels through matrix operations [17], which is in line with people's common sense of understanding images. When people perceive images, they are layered and abstract. Firstly, they understand color and brightness, then local details such as edges, corners, and lines, and then more complex information and structures such as textures and geometric shapes. Finally form the concept of the whole object.

A typical convolutional neural network usually consists of the following three-layer structures: Convolution, Pooling and Fully connected.

2.3.1 Convolution

The convolution layer uses the convolution kernel to gradually scan the image matrix while sliding, calculates the product of the image matrix and the convolution kernel and sums them up, and finally obtains the output pixel matrix. The general feature extraction formula of the convolution layer is shown as follows:

$$X_j^l = f\left(\sum_{i \in M_j} X_i^{l-1} \times k_{ij}^l + b_j^l\right) \quad \textbf{(1)}$$

2.3.2 Pooling

In order to further reduce the dimension of the feature image obtained from the convolution layer, the pooling layer, also known as the downsampling layer, replaces a part of the image with a value. Common methods are Max Pooling and Average Pooling. While Max Pooling is applied to linear functions, Average Pooling is applied to nonlinear functions. In general, Max pooling has better processing effect. The pooling layer has the characteristics of preventing overfitting, improving calculation speed and robustness, and being insensitive to translation and rotation. It is reflected in the recent research on driver fatigue state recognition based on deep learning and information fusion [18].

2.3.3 Fully Connected

After multiple pooling and convolution, the last step is generally to go through the fully connected layer. The function of the fully connected layer is to map the network features to the label space of the sample for prediction, which is actually to achieve the final classification. All the features of things are divided into different categories, so that when a new image is input, the prediction of the new image can be realized according to the classification. The fully connected layer is usually connected after the last convolutional layer. In actual training, the fully connected layer is also implemented by convolution operation.

2.3.4 Activation Function

The activation function is also known as the loss function. The common loss function is Zero-one Loss. The principle is that the same as the predicted value is 0, and the difference is 1. The formula is:

$$L(y_i, f(x_i)) = \begin{cases} 0 & if\ y_i = f(x_i) \\ 1 & if\ y_i \neq f(x_i) \end{cases} \quad \textbf{(2)}$$

The Hinge Loss principle is mainly used in supporting vector machines (SVM). If the classification is correct, the loss value is 0, and if it is wrong, it is 1-f(x). Hinge loss is used to solve the interval maximization problem in SVM. Its formula is as follows:

$$l(f(x), y) = \max(0, 1 - yf(x)) \quad \textbf{(3)}$$

Finally, the Softmax function is used as the output layer. The Softmax function receives a k-dimensional vector as input, and then converts the value of each dimension into a value in the interval (0, 1). Assuming that the input of the network is Y1, Y2,⋯, Yk, the operation of the Softmax function is shown as follows:

$$P(Y = i \mid x) = \text{softmax}(Y_i) = \frac{e^{\omega_i Y_i}}{\sum_{k=1}^{K} e^{\omega_k Y_k}} \quad \textbf{(4)}$$

Among them, P represents the probability that the sample vector x belongs to the ith category, K represents the total number of categories, and ω represents the weight item. The result of adding these probability values is 1, and the final output is that the input visual information belongs to the category to which the maximum value of the 7 expression probabilities belongs [19].

2.4 Lightweight Neural Network (MobileNetV2)

2.4.1 Depthwise Separable Convolution

Bottleneck in MobileNet uses inverted residuals and has been shown to help improve accuracy, so MobileNetV2 also introduces similar blocks. Its core idea is depthwise separable convolution, and the MobileNet has a smaller volume, less computation, and higher accuracy. It has great advantages in lightweight neural networks.

There are two forms of depthwise separable convolution. One is the depthwise convolution, the other is the pointwise convolution. A convolution kernel is only responsible for one channel by depthwise convolution, and a channel is convolved with only one convolution kernel. The number of channels generated by this process is exactly the same as the number of channels in the input. For example, for a 5x5x3 input image, after depth-by-depth convolution, because DW is completely calculated in a two-dimensional plane, the channels and convolution kernels should correspond one-to-one. Therefore, the three-channel input image will correspond to three feature maps. Owing to the number of feature map generated by depthwise convolution is the same as the number of channels in the input layer, the number of channels cannot be compressed. In addition, it performs independent operations on the input layer, so there is no effective use of the same space information on different channels. Therefore, point-by-point convolution came into being to solve the above deficiencies. Point-by-point convolution is very similar to conventional convolution. In fact, it is a standard convolution with a convolution kernel size of 1. The number of channels in the convolution kernel is the same for each layer. The convolution operation will affect the depth-by-depth convolution. The resulting feature maps are weighted and combined in the depth direction to generate a new feature map, and the number of output feature maps is consistent with the number of convolution kernels.

2.4.2 Inverted Residuals

Because Mobilenet uses a deep convolutional neural network, which limits the dimension of input features, resulting in the reduction of features when processing with traditional residual blocks. Therefore, a method of inverted residuals is introduced here. Firstly, the feature map features are expanded, then depth-by-depth and point-by-point convolutions are performed, and finally linear transformation is performed to reduce the dimension. The general process is to expand the 1x1 dimension to 3x3, and perform depth convolution and ReLU, and then is linearly transformed and reduced the dimension to 1x1.

2.4.3 Network Structure

The structure of MobilenetV2 is composed of multiple Bottleneck, Convolution, Pooling, fully connected. Bottleneck adopts the idea of an inverted residual, as shown in the Figure 4.

The ReLU activation function uses ReLU6. The formula is shown as follows:

$$\mathrm{Re}\,LU6 = \min(6, \max(0, x)) \quad \textbf{(5)}$$

To put it bluntly, ReLU6 is an ordinary ReLU but the maximum output value is limited to 6. Owing to ReLU is a nonlinear activation function, when the accuracy is low, it will cause the loss and destruction of information. Therefore, the linear activation function is used here.

Regarding the explanation of the main parameters in the experiment, we adopt adam, an adaptive learning rate method, as the model training optimizer during the model training phase. Following the paper [20], we use initial learning rate of 0.045, and learning rate decay rate of 0.98. The loss function of the model is categorical_crossentropy. When the fully connected layer of the model is mapped to the final classification, softmax is used as the activation function, which is usually combined with the loss function categorical_crossentropy for multi-classification problems. After testing with different values, it is reasonable to believe that when the batch_size is 16, MobileNet converges faster and the effect is better. Alpha, which floats between 0 and 1, is a parameter that controls the width of the network. It is usually set to 1. In terms of model training weights, the weights pre-trained on ImageNet are used to initialize our model.

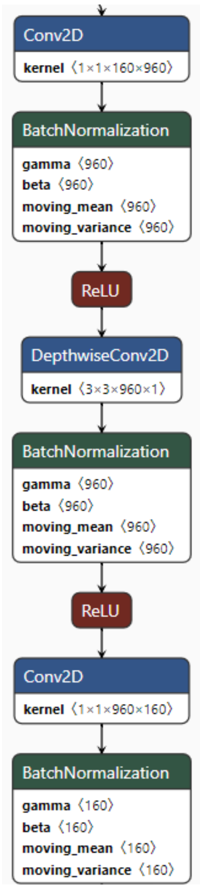

Figure 4. Part of the structure of the model

2.5 Experimental Model

In the experiment, the six neural network models - MobileNet, AlexNet, ResNet, VGG16, VGG19 and YOLOv5 - are applied to the same dataset for comparison.

2.5.1 AlexNet

The AlexNet constructed in this study is divided into 8 layers, and its detailed analysis is shown in the Table 2.

2.5.2 ResNet

The ResNet constructed in this study is divided into 7 layers, and its detailed analysis is shown in the Table 3.

2.5.3 VGG16/VGG19

The VGG16 and VGG19 constructed in this study are divided into 8 layers, and their detailed analysis is shown in the Table 4.

Table 2. Layer analysis of AlexNet

Layer	Detailed analysis of layers
Layer 1	The size of the convolution kernel is 11×11×3, the number is 96, the padding is 0, the stride is 4, and the size of the output image is 55×55×96. The pooling layer adopts Max pooling, and the size of the pooling unit is set to 3×3. Its Padding is 0, Stride is 2, and the size of the pooled image becomes 27×27×96 (consisting of two sets of images of size 27×27×48).
Layer 2	The size of the convolution kernel is 5×5×96, the number is 256, the padding is 2, the stride is 1, and the size of the output image is 27×27×256. The pooling layer adopts Max pooling, and the size of the pooling unit is set to 3×3. Its Padding is 0, Stride is 2, and the size of the pooled image becomes 13×13×256 (consisting of two sets of images of size 13×13×128).
Layer 3&4	The convolution kernel size is 3×3, the number is 384, Padding is 1, Stride is 1, and the output size becomes 13×13×384.
Layer 5	The size of the convolution kernel is 3×3×384, the number is 256, the padding is 1, the stride is 1, and the size of the output image is 13×13×256. The pooling layer adopts Max pooling, and the size of the pooling unit is set to 3×3. Its Padding is 0, Stride is 2, and the size of the pooled image becomes 6×6×256 (consisting of two sets of images of size 6×6×128).
Layer 6&7&8	Fully connected layer. The size of the first two layers is 1×1×4096, and the size of Layer 8 is 1×1×1000. ReLU is used as the activation function. Finally, the result is output through the softmax function.

Table 3. Layer analysis of ResNet

Layer	Detailed analysis of layers
Layer 1	The size of the convolution kernel is 7×7, the number is 64, the padding is 3, the stride is 2, and the size of the output image is 64×112×112. The pooling layer adopts Max pooling, and the size of the pooling unit is set to 3×3. The size of the pooled image becomes 64×56×56.
Layer 2	First, reduce the dimension through the convolution kernel of size 1×1, then increase the dimension through the convolution kernel of size 3×3, and finally reduce the dimension through the convolution kernel of size 1×1, and the output size becomes 256 ×56×56. This step needs to be repeated three times.
Layer 3	First, reduce the dimension through the convolution kernel of size 1×1, then increase the dimension through the convolution kernel of size 3×3, and finally reduce the dimension through the convolution kernel of size 1×1, and the output size becomes 512 ×28×28. This step needs to be repeated four times.
Layer 4	First, reduce the dimension through the convolution kernel of size 1×1, then increase the dimension through the convolution kernel of size 3×3, and finally reduce the dimension through the convolution kernel of size 1×1, and the output size becomes 1024×14×14. This step needs to be repeated six times.
Layer 5	First, reduce the dimension through the convolution kernel of size 1×1, then increase the dimension through the convolution kernel of size 3×3, and finally reduce the dimension through the convolution kernel of size 1×1, and the output size becomes 2048×7×7. This step needs to be repeated three times.
Layer 6&7	Fully connected layer. Using the Average Pool method, the final output size is 2048×1×1.

Table 4. Layer analysis of VGG16

Layer	Detailed analysis of layers
Layer 1	The size of the convolution kernel is 3 × 3, and the number is 64. Two convolutions are required, and ReLU is used as the activation function, and the size of the output image is 224 × 224 × 64. The pooling layer adopts Max pooling, and its pooling unit size is 2×2. The size of the image after pooling becomes 112×112×64.
Layer 2	The size of the convolution kernel is 3 × 3, and the number is 128. Two convolutions are required, and ReLU is used as the activation function, and the size of the output image is 112× 112× 128. The pooling layer adopts Max pooling, and its pooling unit size is 2×2. The size of the image after pooling becomes 56×56×128.
Layer 3	The size of the convolution kernel is 3 × 3, and the number is 256. Three (four for VGG19) convolutions are required, and ReLU is used as the activation function, and the size of the output image is 56× 56× 256. The pooling layer adopts Max pooling, and its pooling unit size is 2×2. The size of the image after pooling becomes 28×28×256.
Layer 4	The size of the convolution kernel is 3 × 3, and the number is 512. Three (four for VGG19) convolutions are required, and ReLU is used as the activation function, and the size of the output image is 28× 28× 512. The pooling layer adopts Max pooling, and its pooling unit size is 2×2. The size of the image after pooling becomes 14×14×512.
Layer 5	The size of the convolution kernel is 3 × 3, and the number is 512. Three (four for VGG19) convolutions are required, and ReLU is used as the activation function, and the size of the output image is 14× 14× 512. The pooling layer adopts Max pooling, and its pooling unit size is 2×2. The size of the image after pooling becomes 7×7×512.
Layer 6&7&8	Fully connected layer. The size of the first two layers is 1×1×4096, and the size of Layer 8 is 1×1×1000. ReLU is used as the activation function. Finally, the result is output through the softmax function.

2.5.4 YOLOv5

The structure of YOLOv5 is shown in Figure 5. The training of the YOLOv5 model adopts the pre-training method, based on the YOLOv5 trained from a large data set, which plays a supporting role for us to train our own dataset. This paper used GPU for training, read in every 4 images as a group, and set the training times to 100. We then applied it to our own dataset, resulting in a model for forest fires.

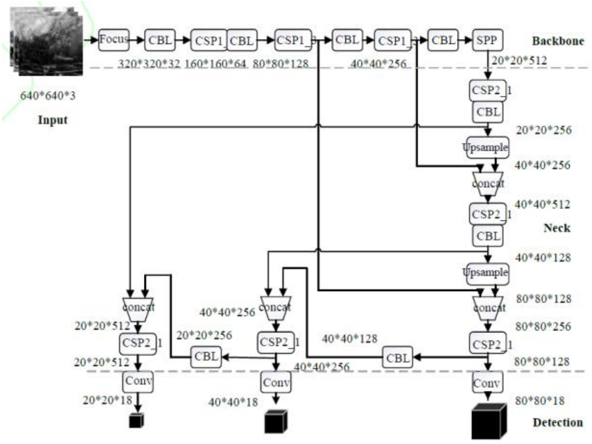

Figure 5. The structure of YOLOv5

2.6 Experimental Results

The six models were tested on the same test data respectively, and the corresponding accuracy and Average Prediction Time (APT) were obtained. APT is used to measure the average time loss of predicting each image. The test data is a total of 1071 images including three subsets of Fire, Smoke and Fog. The accuracy and APT of the six models are shown in Table 5. The accuracy and loss of models are shown in Figure 6 and Figure 7 respectively. The confusion matrix diagram of YOLOv5 and prediction of results for YOLOv5 are shown in Figure 8 and Figure 9 respectively.

Table 5. Evaluation of models

Model	Accuracy (%)	APT (s/image)
MobileNet	95.8	0.02109
AlexNet	88.7	0.02821
ResNet	92.3	0.02247
YOLOv5	89.6	0.03084
VGG16	92.1	0.02276
VGG19	91.7	0.02629

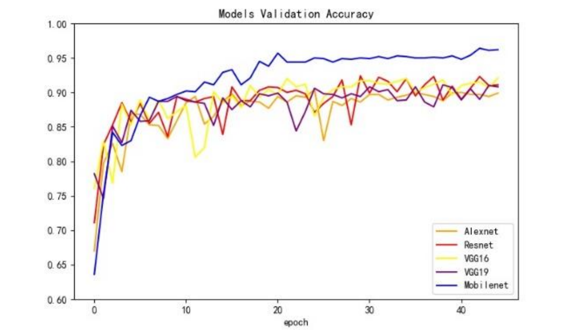

Figure 6. Accuracy of models

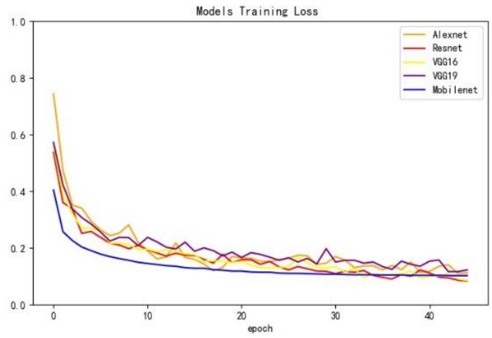

Figure 7. Loss of models

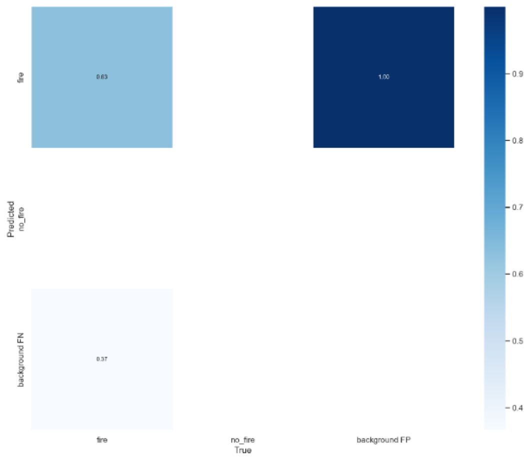

Figure 8. Confusion matrix diagram of YOLOv5

Figure 9. Prediction of results for YOLOv5

The accuracy of the six models are: MobileNet is 95.8%; AlexNet is 88.7%; ResNet is 92.3%; YOLOv5 is 89.6%; VGG16 is 92.1%; VGG19 is 91.7%. MobileNet. The APT of the 6 models are: MobileNet is 0.02109; AlexNet is 0.02821; ResNet is 0.02247; YOLOv5 is 0.03084; VGG16 is 0.02276; VGG19 is 0.02629. Combining Figure 6 and Figure 7, we can observe that MobileNet achieves a relatively good convergence effect before and after the 25th iteration. The accuracy of MobileNet reaches 95.8%, achieving a high recognition accuracy. Considering that MobileNet is a lightweight convolutional neural network, fewer parameters help it get a lower APT of only 0.02109. As shown in Figure 7, the loss of MobileNet is relatively lower than several other models. Therefore, in general, compared with other models, MobileNet has better performance in the field of forest fire recognition

3 Conclusion

This paper constructed the dataset including three subsets of Fire, Smoke, and Fog by downloading data from existing public datasets, and enhanced the model's generalization and expression capabilities through content transformation and scale transformation, which improved the accuracy of the experiment.

The experiment is a comparative experiment. The six models of MobileNet, AlexNet, ResNet, VGG16, VGG19 and YOLOv5 are used to identify the same dataset, so as to compare and determine which model has better recognition efficiency and wider application scenarios. In forest fire recognition, MobileNet significantly reduces the number of parameters, which helps to have a lower APT. At the same time, its recognition accuracy rate is as high as 95.2%, which better improves the recognition performance. In addition, the model itself has a certain degree of scene generalizability, which is not only applicable to forest fires, but also can be extended to indoor fires and flame identification, which provides a new theoretical calculation model for fire prevention.

Acknowledgements

Supported by the Innovation and Entrepreneurship Training Program of Guangdong Ocean University (No. CXXL2021243); Program for Scientific Research Start-up Funds of Guangdong Ocean University (No. R20079).

References

[1] S. Chen, *Experimental Study on the Effects of Fuel Load and Packing Ratio on Fire Spread under Zero-Slope Condition*, Master Thesis, University of Science and Technology of China, Hefei, China, 2020.

[2] H. Huang, W. Wu, Z. Ye, H. Zhang, J. He, Design of Forest Fire Monitoring System based on ZigBee, *Shihezi Science and Technology*, No. 2, pp. 53-54, April, 2021.

[3] Y. Qin, R. Zhou, K. Zhang, X. Fan, R. Feng, Forest Fire Monitoring and Early Warning System based on NB-IoT Narrowband Communication and Multi-sensor Networking Technology, *Internet of Things Technologies*, Vol. 10, No. 6, pp. 14-16+19, June, 2020.

[4] W. Wang, X. He, S. Zhang, Study on Multi-source Information Forest Fire Monitoring Model Based on Radar, *Advances in Meteorological Science and Technology*, Vol. 10, No. 6, pp. 101-105, December, 2020.

[5] J. Yang, F. Liu, T. Zhang, Q. Zhang, Y. Ren, Smoke Image Recognition Based on Improved Residual Network, *Science Technology and Engineering*, Vol. 19, No. 32, pp. 236-243, November, 2019.

[6] J. Du, Q. Chang, X. Liu, DBN-CNN for forest fire smoke recognition, *Modern Electronics Technique*, Vol. 43, No. 13, pp. 44-48, July, 2020.

[7] L. Zhu, Forest Smoke Detection Algorithm based on Image Enhancement and Multi-feature of texture, *Fire Science and Technology*, Vol. 37, No. 2, pp. 225-228, February, 2018.

[8] J. Li, R. Fan, Z. Chen, Forest Fire Recognition based on Color and Texture Features, *Journal of South China University of Technology (Natural Science Edition)*, Vol. 48, No. 1, pp. 70-83, January, 2020.

[9] Z. Li, Y. Chen, Y. Song, K. Lu, J. Shen, Effective Covering Array Generation Using an Improved Particle Swarm Optimization, *IEEE Transactions on Reliability*, Vol. 71, No. 1, pp. 284-294, March, 2022.

[10] F. Zhou, L. Jin, J. Dong, Review of Convolutional Neural Network, *Chinese Journal of Computers*, Vol. 40, No. 6, pp. 1229-1251, January, 2017.

[11] S. S. Patil, S. S. Pardeshi, N. Pradhan, A. D. Patange, Cutting Tool Condition Monitoring using a Deep Learning-based Artificial Neural Network, *International Journal of Performability Engineering*, Vol. 18, No. 1, pp. 37-46, January, 2022.

[12] D. Li, W. E. Wong, W. Wang, Y. Yao, M. Chau, Detection and Mitigation of Label-flipping Attacks in Federated Learning Systems with KPCA and K-means, *2021 8th International Conference on Dependable Systems and Their Applications (DSA)*, IEEE, Yinchuan, China, 2021, pp. 551-559.

[13] Y.-F. Chen, Y.-K. Lin, C.-F. Huang, Using Deep Neural Networks to Evaluate the System Reliability of Manufacturing Networks, *International Journal of Performability Engineering*, Vol. 17, No. 7, pp. 600-608, July, 2021.

[14] W. Liu, J. Li, H. Qu, Research on Traffic Sign Recognition based on Multi-scale Convolutional Neural Network, *Application Research of Computers*, pp. 1-7, December, 2021.

[15] W. Wu, Y. Li, Emfacenet: A Convolutional Neural Network for Lightweight Face Recognition, *Journal of Chinese Computer Systems*, pp. 1-6, December, 2021.

[16] Y. Yue, B. Sun, H. Wang, H. Zhao, Object Detection of Tomato Fruit Based on Cascade RCNN, *Science Technology and Engineering*, Vol. 21, No. 6, pp. 2387-2391, February, 2021.

[17] C. Megha, S. M. Kumar, R. K. Kumar, Intelligent Optimization of Latent Fingerprint Image Segmentation using Stacked Convolutional Autoencoder, *International Journal of Performability Engineering*, Vol. 17, No. 4, pp. 379-393, April, 2021.

[18] J. Bian, *Research on Multi-source Heterogeneous Adaptive Integrated Navigation Technology in Complex Environment*, Master Thesis, Shandong University, Jinan, China, 2020.

[19] P. Zhang, W. Kong, J. Teng, Facial Expression Recognition based on Multi-scale Feature Attention Mechanism, *Computer Engineering and Applications*, Vol. 58, No. 1, pp. 182-189, January, 2022.

[20] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, MobileNetV2: Inverted Residuals and Linear Bottlenecks, *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, 2018, pp. 4510-4520.

Biographies

Zhixiang Li is an undergraduate student at the College of Mathematics and Computer, Guangdong Ocean University, Zhanjiang, majoring in computer science and technology. His research interests are machine learning and neural networks.

Hongbin Jiang is currently studying computer science at Guangdong Ocean University. His research focuses on computer vision and digital image processing.

Qixiang Mei, PhD, is an associate professor of computer science and technology at Guangdong Ocean University. His research focuses on big data processing, neural network and information security, etc. He presided over a project of the National Natural Science Foundation of China.

Zhao Li received the MS and PhD degrees in computer science and technology from Wuhan University. He is a professor in Computer Science and Technology, Guangdong Ocean University. His research focuses on image recognition, big data processing, and software defects prediction.